MA35D1
GUI Development Environment

Agenda

- DirectFB Introduction
- Qt Creator Cross-Compile Environment Setup

DirectFB Introduction

DirectFB Introduction

- Nuvoton provides DirectFB, a 2D graphics engine that can offload CPU usage and improve application performance.
- The amount of CPU usage saved depends on the content of the applications being run

Test item	CPU loading
QT with DirectFB and 2D	41%
QT with DirectFB without 2D	86%
QT without DirectFB and 2D	78%

- GC520L (2D graphic engine) shared library is located at /usr/lib/directfb-1.7-7/gfxdrivers/libdirectfb_gal.so
- Remove libdirectfb_gal.so and run the demo will use DirectFB without 2D Engine
- DirectFB can be configured by creating *directfbrc* configuration file at /etc

Qt Creator Cross-Compile Environment Setup

Build Toolchain from Yocto

- First, you need to build ma35d1 toolchain from Yocto.
- Create a toolchain installer. If you have already created one, do not execute the command again.

```
$ bitbake nvt-image-qt5 -c populate_sdk
```

- After build toolchain SDK, it will locate at /tmp-glibc/deploy/sdk
- Use host OS Ubuntu to install it to the default path

```
$ ./oecore-x86_64-aarch64-toolchain-5.5-dunfell.sh
```

```
user@ubuntu:~/sdk$ sudo ./oecore-x86_64-aarch64-toolchain-5.5-dunfell.sh
[sudo] password for user:
Nuvoton Release Distro SDK installer version 5.5-dunfell
==========================================================
Enter target directory for SDK (default: /usr/local/oecore-x86_64):
You are about to install the SDK to "/usr/local/oecore-x86_64". Proceed [Y/n]? y
Extracting SDK.....................................................................................
.....................................................done
Setting it up...done
SDK has been successfully set up and is ready to be used.
Each time you wish to use the SDK in a new shell session, you need to source the environment setup script e.g.
 $ . /usr/local/oecore-x86_64/environment-setup-aarch64-poky-linux
```

Toolchain from Buildroot

- Buildroot have already installed the toolchain

~/Buildroot/MA35D1_Buildroot/output/host/usr/bin$

```
user@ubuntu:~/Buildroot/MA35D1_Buildroot/output/host/usr/bin$ ls
2to3                                aarch64-nuvoton-linux-gnu-as                autom4te                fdtput                  icuinfo     mcookie
2to3-3.9                            aarch64-nuvoton-linux-gnu-c++               automake                fincore                 ifnames     mcopy
aarch64-linux-addr2line             aarch64-nuvoton-linux-gnu-c++.br_real       automake-1.15           findmnt                 infocmp     mdel
aarch64-linux-ar                    aarch64-nuvoton-linux-gnu-cc                autopoint               fixqt4headers.pl        infotocap   mdeltree
aarch64-linux-as                    aarch64-nuvoton-linux-gnu-cc.br_real        autoreconf              flex                    ipcmk       mdir
aarch64-linux-c++                   aarch64-nuvoton-linux-gnu-c++filt           autoscan                flex++                  ipcrm       mdu
aarch64-linux-c++.br_real           aarch64-nuvoton-linux-gnu-cpp               autoupdate              flock                   ipcs        meson
aarch64-linux-cc                    aarch64-nuvoton-linux-gnu-cpp.br_real       awk                     floppyd                 isosize     mformat
aarch64-linux-cc.br_real            aarch64-nuvoton-linux-gnu-elfedit           bison                   floppyd_installtest     jq          minfo
aarch64-linux-c++filt               aarch64-nuvoton-linux-gnu-g++               cal                     gapplication            kill        mk_cmds
aarch64-linux-cpp                   aarch64-nuvoton-linux-gnu-g++.br_real       captoinfo               gawk                    kmod        mkenvimage
aarch64-linux-cpp.br_real           aarch64-nuvoton-linux-gnu-gcc               chacl                   gdbus                   last        mkimage
aarch64-linux-elfedit               aarch64-nuvoton-linux-gnu-gcc-9.4.0         chattr                  gdbus-codegen           lastb       mkmanifest
aarch64-linux-g++                   aarch64-nuvoton-linux-gnu-gcc-9.4.0.br_real  choom                   genbrk                  libtool     mkpasswd
aarch64-linux-g++.br_real           aarch64-nuvoton-linux-gnu-gcc-ar            clear                   gencfu                  libtoolize  mlabel
aarch64-linux-gcc                   aarch64-nuvoton-linux-gnu-gcc.br_real       cmake                   gencnval                linux32     mmd
aarch64-linux-gcc-9.4.0             aarch64-nuvoton-linux-gnu-gcc-nm            col                     gendict                 linux64     mmount
aarch64-linux-gcc-9.4.0.br_real     aarch64-nuvoton-linux-gnu-gcc-ranlib        colcrt                  genimage                linux-dtc   mmove
aarch64-linux-gcc-ar                aarch64-nuvoton-linux-gnu-gcov              colrm                   genrb                   logger      moc
aarch64-linux-gcc.br_real           aarch64-nuvoton-linux-gnu-gcov-dump          column                  getfacl                 look        mount
aarch64-linux-gcc-nm                aarch64-nuvoton-linux-gnu-gcov-tool         compile_et              getfattr                lsattr      mountpoint
aarch64-linux-gcc-ranlib            aarch64-nuvoton-linux-gnu-gdb               convert-dtsv0           getopt                  lsblk       mpartition
```

qmake from Buildroot

- Because Buildroot compile qmake in the docker, you need to create a “shared” folder soft link to meet the path about qmake

```
$ ln –s ./ shared
```

```
user@ubuntu:~$ ln -s ./ shared
user@ubuntu:~$ ls
Buildroot  Docker     MA35D1_NuWriter  Pictures  Qt_Project  Videos
config     Documents  Music            Public    shared      yocto
Desktop    Downloads  patch            Qt5.14.1  Templates
```

Download and Install Qt Creator

- Download QT Creator from https://download.qt.io/archive/qt/5.14/5.14.1/ under host Ubuntu operating environment. Choose qt-opensource-linux-x64-5.14.1.run.
- After downloading, create a folder you prepare to install QT Creator under /home/user

```
$ mkdir /home/user/Qt_Creator
```

- Go to the path you downloaded and install QT Creator

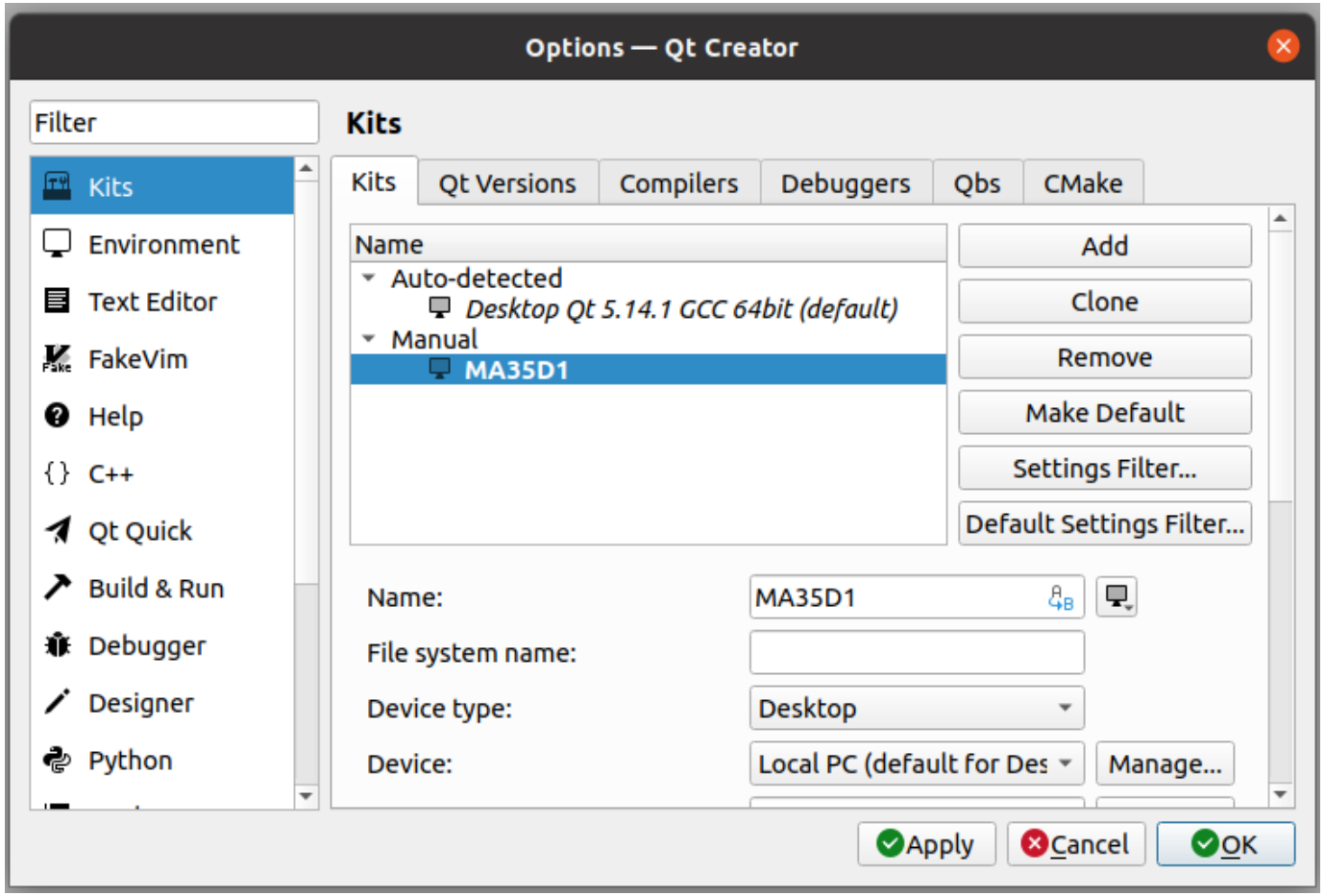
```
$ chmod 777 qt-opensource-linux-x64-5.14.1.run
$ sudo ./qt-opensource-linux-x64-5.14.1.run
```

- After installed, open QT Creator with the command below

```
user@ubuntu:~/QT_Creator/Tools/QtCreator/bin$ ./qtcreator.sh &
```

Qt Creator option – Kits

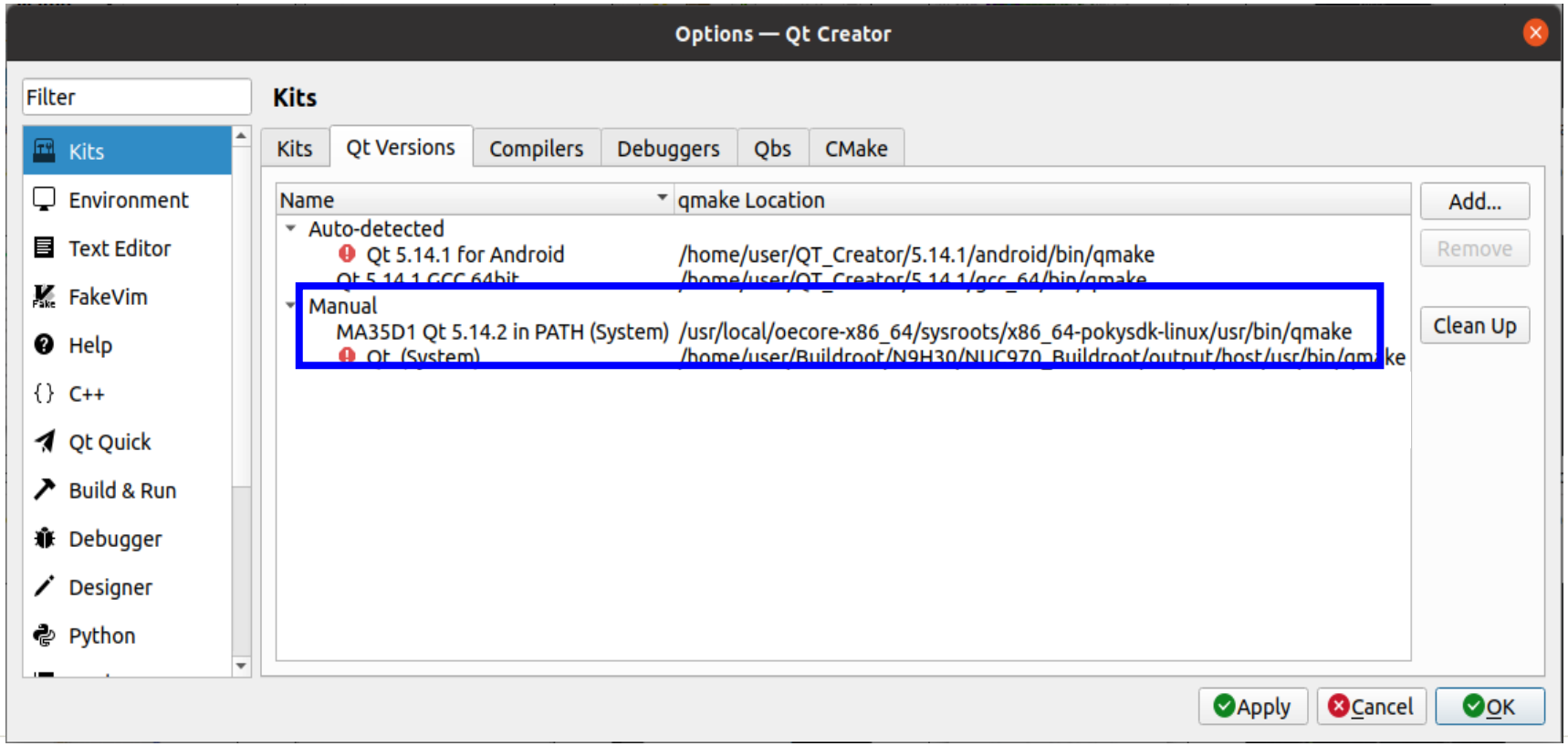
- Create a MA35D1 kits to build the application



NuDeveloper Ecosystem – Make the engineers' job easier.

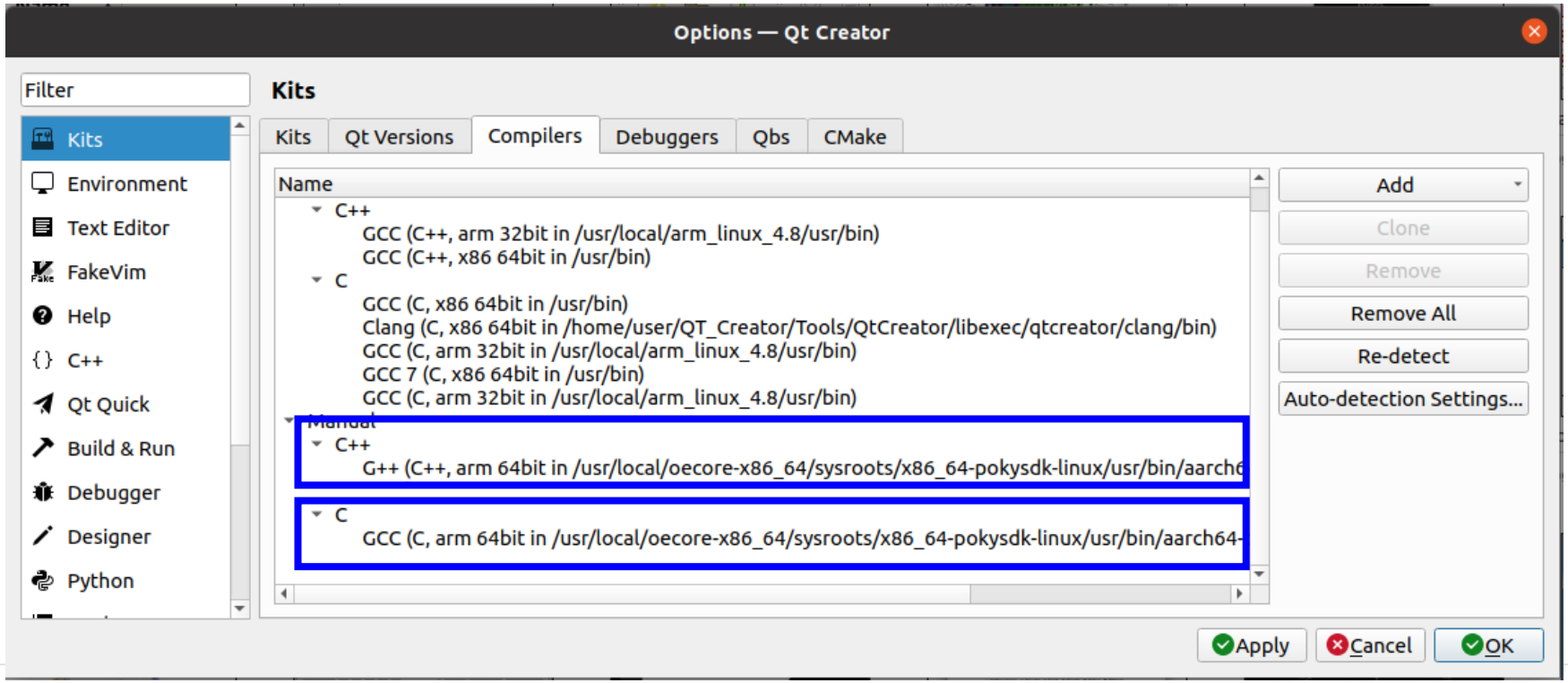
Qt Creator option

- Tools -> Options -> Qt versions need to be match the version you built
  - Buildroot: ~/MA35D1_Buildroot/output/host/bin/qmake
  - Yocto: /usr/local/oecore-x86_64/sysroots/x86_64-pokysdk-linux/usr/bin/qmake



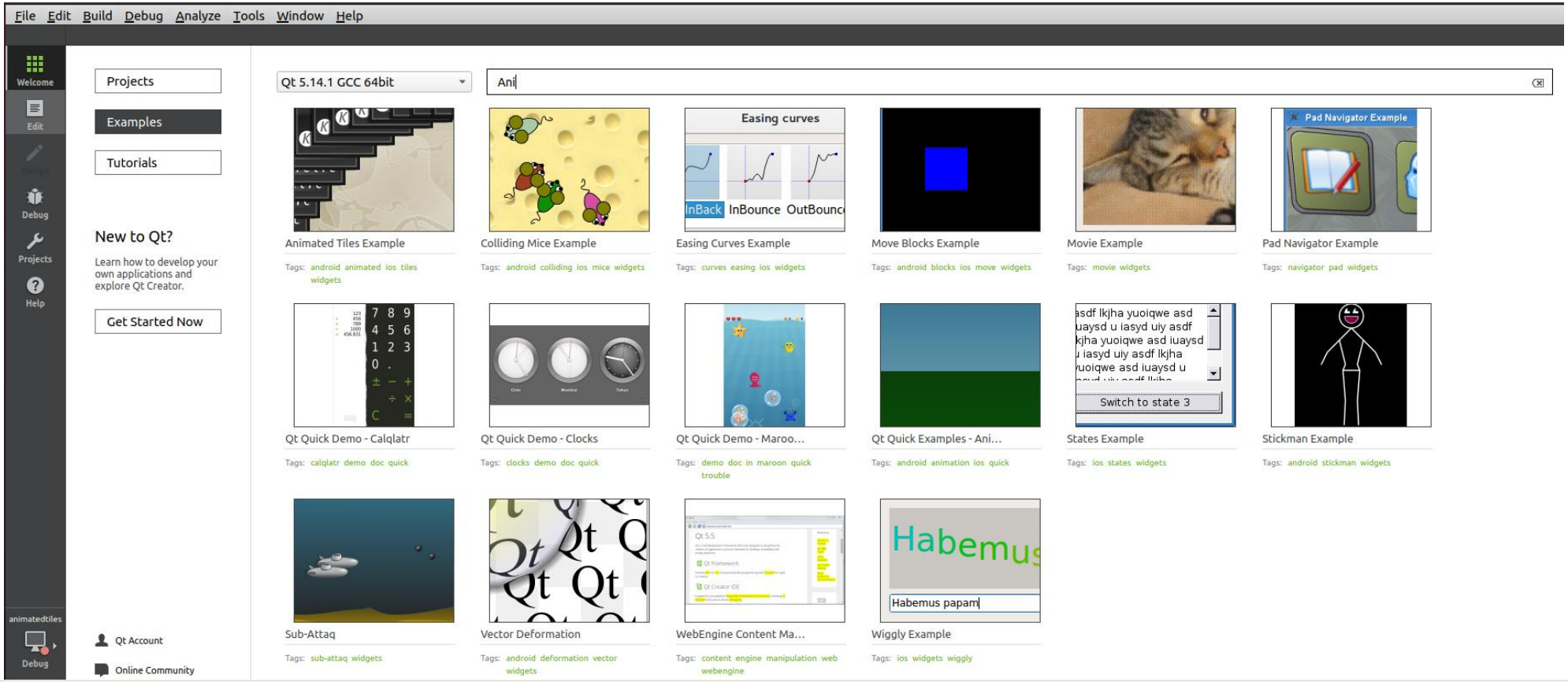
Qt Creator option – Compilers

- Choose the two toolchains
  - Buildroot: ~/MA35D1_Buildroot/output/host/usr/bin/aarch64-linux-gcc
  - Yocto: /usr/local/oecore-x86_64/sysroots/x86_64-pokysdk-linux/usr/bin/aarch64-poky-linux/aarch64-poky-linux-g++



Qt Application Demo Code

- Qt Creator provides many example for users to use
- Choose Animated Tiles

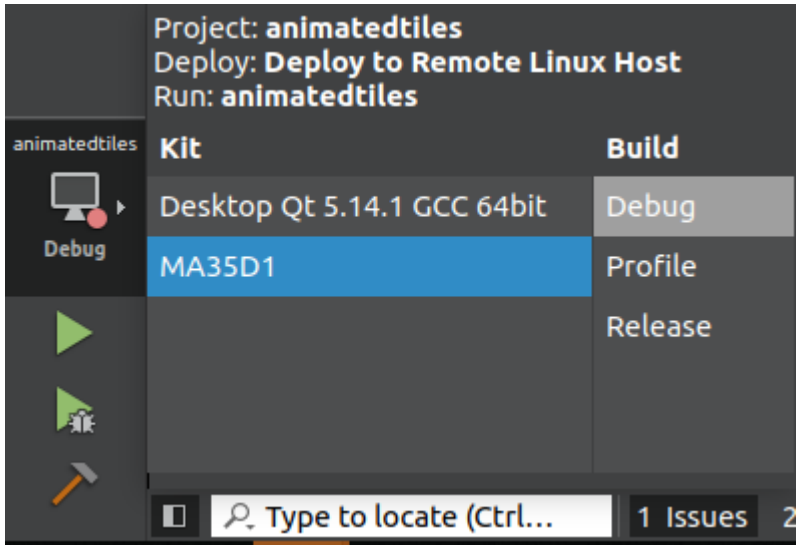


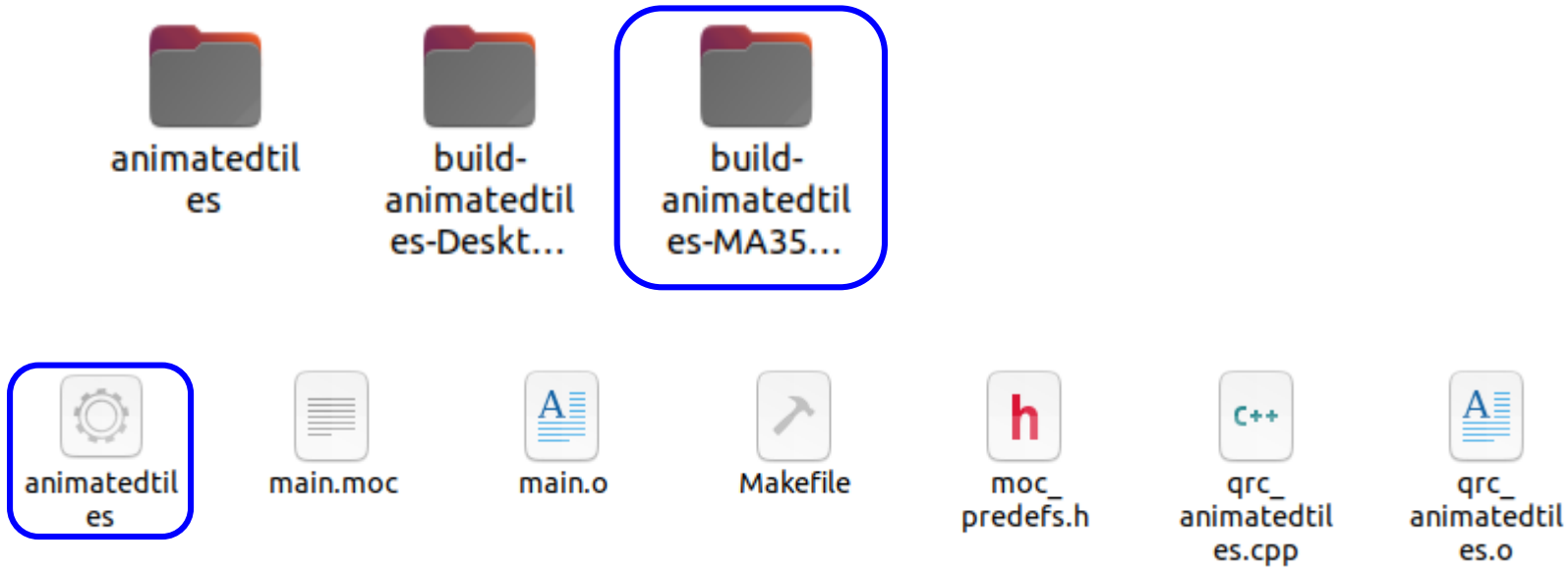
NuDeveloper Ecosystem – Make the engineers' job easier.

nuvoTon

QT Application Demo Code

- There two complier kits can be choose. One can be run directly on a PC, and the other can be run on MA35D1.
- After building, Qt Creator will create a 'Debug' folder located at the path you chose
- Copy it to the evaluation board





Qt and TSLIB Environment Variable

- Before running the Qt application, ensure that the Qt environment variables are properly configured. If you are using Yocto to build the application, there is no need to manually export the environment variables as they are already set up in the Yocto build environment.

```
export QT_QPA_FB_TSLIB="1"
export QT_QPA_FONTDIR="/usr/share/fonts/truetype"
export QT_QPA_PLATFORM="linuxfb"
export TERM="xterm"
export TSLIB_CONSOLEDEVICE=none
export TSLIB_FBDEVICE=/dev/fb0
export TSLIB_TSDEVICE=/dev/input/event0
export TSLIB_CONFFILE=/etc/ts.conf
export TSLIB_PLUGINDIR=/usr/lib/ts
export TSLIB_CALIBFILE=/etc/pointercal
export LD_LIBRARY_PATH=/lib:/usr/lib:/usr/lib/ts
```

Run Qt Application

- Execute the application with linuxfb
- Yocto:

```
root@numaker-som-ma35d16a81:~# /usr/share/qt5everywheredemo-1.0/QtDemo
```

- Buildroot:

```
root@numaker-som-ma35d16a81:~#
/usr/lib/qt/examples/widgets/animation/animatedtiles/animatedtiles
```

- Nuvoton provide 2D Graphic Engine with DirectFB
- Add –platform directfb to the command upper run the application with directfb

Joy of innovation

NUVOTON

谢谢

謝謝

Děkuji

Bedankt

Thank you

Kiitos

Merci

Danke

Grazie

ありがとう

감사합니다

Dziękujemy

Obrigado

Спасибо

Gracias

Teşekkür ederim

Cảm ơn